WHAT IS CLAIMED IS:

1. A method of verifying whether a specified computer program satisfies a predefined set of conditions, comprising:

converting the program into a logical equation representing the predefined set of conditions as applied to instructions and elements of the instructions of the program, the converting step including inserting flow control labels into the sub-equations of the logical equation, the flow control labels identifying conditional branch points in the specified computer program;

applying a theorem prover to the logical equation to determine the truth of the logical equation, and when the truth of the logical equation cannot be proved, generating at least one counter-example identifying one of the conditions, one or more variable values inconsistent with the one condition, and any of the flow control labels for conditional branch points of the program associated with the identified variable values; and

converting the at least one counter-example into an error message that includes a program trace when the counter-example identifies one or more of the flow control labels.

2. The method of claim 1 wherein the converting step additionally comprises a step of converting the program into an intermediate language form prior to creating the logical equation.

3. The method of claim 2 wherein the flow control labels are inserted before converting the program into the intermediate language form.

4. The method of claim 2 wherein the flow control labels are inserted after converting the program into the intermediate language form.

5. The method of claim 2 wherein the intermediate language form is Dijkstra's guarded command language.

6. The method of claim 1 wherein at least one of the flow control labels includes a flow control label name that includes a string that identifies a type of branch in the program and a line number in the specified computer program.

7. The method of claim 1 wherein at least one of the flow control labels includes a flow control label name that includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

8. The method of claim 1 wherein at least one of the flow control labels is of the form L ==> P wherein L is a flow control label name and P is a subcomponent of the logical equation.

9. The method of claim 8 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

10. The method of claim 8 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

11. The method of claim 1 wherein at least one of the flow control labels is of the form L=k ==> P wherein L is a flow control label name, k is a constant value and P is a subcomponent of the logical equation.

12. The method of claim 11 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

13. The method of claim 11 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

14. The method of claim 1 wherein at least one of the flow control labels is of the form {LBLPOS L P} wherein L is a flow control label name and P is a subcomponent of the logical equation.

15. The method of claim 14 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

16. The method of claim 14 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

17. The method of claim 1 wherein at least one of the flow control labels is of the form ¬{LBLNEG L ¬P} wherein L is a flow control label name and P is a subcomponent of the logical equation.

18. The method of claim 17 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

19. The method of claim 17 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

20. The method of claim 1 wherein at least one of the flow control labels is of the form {LBLPOS L True} ==> P wherein L is a flow control label name and P is a subcomponent of the logical equation.

21. The method of claim 20 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

22. The method of claim 1 wherein the flow control label name identifies a line number in the specified computer program at which an associated program instruction is located and includes a sequence number indicating an order of execution of the program instruction at the identified line number relative to other program instructions identified by other flow control labels.

23. A computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a verification condition generator that converts a specified program into a logical equation representing the predefined set of conditions as applied to instructions and elements of the instructions of the program, the verification condition generator including instructions that insert flow control labels into the sub-equations of the logical equation, the flow control labels identifying conditional branch points in the specified computer program;

a theorem prover that processes the logical equation to determine the truth of the logical equation, and when the truth of the logical equation cannot be proved, generates at least one counter-example identifying one of the conditions, one or more variable values inconsistent with the one condition, and any of the flow control labels for conditional branch points of the program associated with the identified variable values; and

a post processing module that converts the at least one counter-example into an error message that includes a program trace when the counter-example identifies one or more of the flow control labels.

24. The computer program product of claim 23 wherein the verification condition generator includes instructions for converting the program into an intermediate language form prior to creating the logical equation.

25. The computer program product of claim 24 wherein the verification condition generator includes instructions for inserting the flow control labels before converting the program into the intermediate language form.

26. The computer program product of claim 24 wherein the verification condition generator includes instructions for inserting the flow control labels after converting the program into the intermediate language form.

27. The computer program product of claim 24 wherein the intermediate language form is Dijkstra's guarded command language.

28. The computer program product of claim 23 wherein at least one of the flow control labels includes a flow control label name that includes a string that identifies a type of branch in the program and a line number in the specified computer program.

29. The computer program product of claim 23 wherein at least one of the flow control labels includes a flow control label name that includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

30. The computer program product of claim 23 wherein at least one of the flow control labels is of the form L ==> P wherein L is a flow control label name and P is a subcomponent of the logical equation.

31. The computer program product of claim 30 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

32. The computer program product of claim 30 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

33. The computer program product of claim 23 wherein at least one of the flow control labels is of the form L=k ==> P wherein L is a flow control label name, k is a constant value and P is a subcomponent of the logical equation.

34. The computer program product of claim 23 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

35. The computer program product of claim 23 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

36. The computer program product of claim 23 wherein at least one of the flow control labels is of the form {LBLPOS L P} wherein L is a flow control label name and P is a subcomponent of the logical equation.

37. The computer program product of claim 36 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

38. The computer program product of claim 36 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

39. The computer program product of claim 23 wherein at least one of the flow control labels is of the form ¬{LBLNEG L ¬P} wherein L is a flow control label name and P is a subcomponent of the logical equation.

40. The computer program product of claim 39 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

41. The computer program product of claim 39 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the program and a line number in the specified computer program.

42. The computer program product of claim 23 wherein at least one of the flow control labels is of the form {LBLPOS L True} ==> P wherein L is a flow control label name and P is a subcomponent of the logical equation.

43. The computer program product of claim 42 wherein the flow control label name comprises a string that identifies a type of branch in the program and a line number in the specified computer program.

44. The computer program product of claim 42 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the specified computer program and a line number in the specified computer program.

45. The computer program product of claim 23 wherein the flow control label name identifies a line number in the specified computer program at which an associated program instruction is located and includes a sequence number indicating an order of execution of the program instruction at the identified line number relative to other program instructions identified by other flow control labels.

46. The method of claim 20 wherein the flow control label name includes a value associated with an entry in a table that identifies a type of branch in the specified computer program and a line number in the specified computer program.